

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Inventor: **Gabriel G. Montero, Srinivas Hasti and David B. Colasurdo**

Application No.: **09/823,120**

Examiner: **Avi M. Gold**

Filed: **March 30, 2001**

Group Art Unit: **2157**

Case: **RSW9-2001-0063-US1**

Title: **METHOD AND APPARATUS FOR ASYNCHRONOUS TIME BASED  
UPDATES OF HTTP SESSIONS**

---

**ELECTRONICALLY FILED**

---

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**Attention: Board of Patent Appeals and Interferences**

**CORRECTED APPELLANTS' BRIEF**

This brief is a corrected appeal brief responsive to the Notice of Non-Compliant Appeal Brief dated June 5, 2007 in this case and is in furtherance of the Notice of Appeal filed on March 15, 2007.

**1) REQUIRED FEE**

The requisite fee of \$500.00 set forth in §41.20(b)(2) was previously submitted in this case in connection with Applicant's Appeal Brief filed June 22, 2005 (in response to which the Examiner issued a new Office Action, thereby removing the application from the appeal process. If the submitted fee is insufficient, the United States Patent and

Trademark Office (hereinafter "Office") is authorized to charged Applicant's Deposit Account No. 19-5425 for any shortfall.

**2) REAL PARTY IN INTEREST**

The present application is assigned to International Business Machines Corporation. International Business Machines Corporation is the real party in interest.

**3) RELATED APPEALS AND INTERFERENCES**

The appellant, assignee, and the legal representatives of both are unaware of any other appeal or interference that will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

**4) STATUS OF CLAIMS**

- a. Claims canceled: None
- b. Claims withdrawn from consideration but not canceled: None
- c. Claims pending: 1-24
- d. Claims allowed: None
- e. Claims merely objected to: None
- f. Claims rejected: 1-24
- g. Appealed Claims: 1-24

Appealed claims 1-24 as currently pending are attached as Appendix A hereto.

**5) STATUS OF AMENDMENTS**

There are no un-entered amendments in this application.

**6) SUMMARY OF CLAIMED SUBJECT MATTER**

The invention claimed in the claims on appeal is best illustrated by Figure 2 of the present application and is best described in detail at page 12, line 10 through page 19, line 6 of the specification. The subject matter of all claims pertains to a method and apparatus for updating a session database that is accessible by multiple servers in a Web environment. Page 8, line 14-15. For instance, Web sites often divide the tasks of servicing requests into a three tier system with a different server or plurality of servers to handle each tier. The first, front end tier may be the http servers that process the http aspects of a transaction. The second tier may be the application servers, which handle the content specific processing for the transactions. The third tier may comprise database servers. Within each tier, the Web site server system may have multiple, redundant servers. Since http is a connectionless protocol, one request from a particular client can be directed to one application server while the next request from the same client machine might be directed to a different application server. Accordingly, a means often is provided for the various servers to access the session data developed by another, redundant server.

Commonly, such sharing of http session data is enabled by use of a database server for storing session data that is accessible to the plurality of application servers. Particularly, each application server typically stores its session data in local memory, but also writes a copy of the session data to the session database. If a different server

services a request from a client, that different server can go to the database and read the session data for the corresponding session. The session data is updated in both the local memory and the database each time a request causes a change in the data.

The present invention is designed to reduce the number of writes to an http session database in order to conserve system resources. Page 12, lines 10-11. In accordance with the invention, each server maintains http session data in a local memory. This copy of the http session data will be updated every time there is a change in the session data. Furthermore, the server writes a copy of the session data to the common database shared by all of the servers only at designated times. In a preferred embodiment of the invention, the designated time is periodic. In alternate embodiments, the servers may write the session data to the database (a) after a specified number of requests in that session have been received or (b) after a specified number of changes to the session data have been made. Page 12, line 21 through page 13, line 16 and page 16, lines 8-17.

Below is a copy of the independent claims mapped to their corresponding support in the specification and drawings.

1. A server system utilizing an HttpSession object in a Java servlet application program interface (API) (page 9, lines 14-21) comprising:

a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory (page 12, lines 5-7 and page 13, lines 10-11);

a second memory having a database for storing HttpSession objects for http sessions being handled by said JVMs, said memory being accessible by each of said

JVMs (col. 11, line 1 through page 12, line 9 and page 12, line 21 through page 13, line 16);

a first computer program adapted to store in a memory local to said server running said JVM HttpSession data for each http session handled by said JVM (page 12, lines 5-9, page 16, line 18 to page 17, line 10,m and Fig. 2, left hand side);

a second computer program adapted to write to said database a copy of said HttpSession data for each said http session at a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session (page 12, line 21 through page 13, line 16, page 18, lines 2-22, and Fig. 2, right hand side).

11. A method of maintaining session data in a server system servicing a network, said server system maintaining state data pertaining to sessions (page 9, lines 14-21), said method comprising the steps of:

(1) storing data for each session in a memory local to a server servicing said session (page 12, lines 5-9, page 16, line 18 through page 17, line 10, and Fig. 2, left hand side);

(2) writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server system at designated times, said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions (page 12, line 21 through page 13, line 16, page 18, lines 2-22, and Fig. 2, right hand side).

18. A server system utilizing HttpSession objects in a Java servlet application program interface (API) (page 9, lines 14-21) comprising:

a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory (page 12, lines 5-7 and page 13, lines 10-11);

a memory having a database for storing HttpSession objects for http sessions being handled by said plurality of JVMs, said memory being accessible by each of said JVMs (col. 11, line 1 through page 12, line 9 and page 12, line 21 through page 13, line 16);

a first computer program adapted to store in a memory local to said server running said JVM HttpSession object data for each http session handled by a JVM (page 12, lines 5-9, page 16, line 18 to page 17, line 10, and Fig. 2, left hand side);

a second computer program adapted to write a copy of said HttpSession data for each said http session in said database at designated times, said designated times determined as a function of at least one of (a) the number of times the HttpSession object data is updated in said local memory and (b) the number of times an http request in said http session is serviced (page 12, line 21 through page 13, line 16, page 15, lines 8-17, page 18, lines 2-22, and Fig. 2, right hand side).

## **7) GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL**

1. Claims 11-13, 16, and 17 stand rejected under 35 USC section 103 (a) as unpatentable over U.S. Patent No. 6,480,894 issued to Courts et al. (hereinafter Courts) in view of U.S. Patent No. 7,010,605 issued to Dharmarajan (hereinafter Dharmarajan).

2. Claim 14 stands rejected under 35 U.S.C. §103(a) as unpatentable over Courts and Dharmarajan and further in view of U.S. Patent No. 6,701,438 issued to Prabandham. et al. (hereinafter Prabandham).

3. Claims 1-10, 15, and 18-24 stand rejected under 35 U.S.C. §103(a) as unpatentable over Courts, Dharmarajan, and Prabandham and further in view of U.S. Patent No. 6,745,387 issued to Ng et al. (hereinafter Ng).

## **8) ARGUMENT**

### **A. The Independent Claims 1, 11, and 18**

The application contains three independent claims, namely, claims 1, 11, and 18. The Examiner rejected independent claim 11 as unpatentable over Courts in view of Dharmarajan and independent claims 1 and 18 under 35 U.S.C. 103 (a) as unpatentable over Courts, Dharmarajan, Prabandham and Ng.

Generally, all three independent claims recite a server system having a global database accessible by multiple servers for maintaining copies of the local session data stored locally at each server wherein the global database is updated at intervals other than the conventional interval of every time the local copy of the session data is updated at a particular server.

In independent apparatus claim 1 and independent method claim 11, the interval is claimed as “a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session” (claim 1) and “said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions” (claim 11). Although these two independent claims recite a designated interval in different language, they share the basic premise of the interval being a predetermined time interval since the last write to the global database.

Independent claim 18 recites a different embodiment in which the interval is “determined as a function of at least one of (a) the number of times the HttpSession object data is updated in said local memory and (b) the number of times an http request in said http session is serviced”.

The fundamental dispute between applicant and the Examiner in this case is whether the prior art of record teaches that the claimed intervals for updating the global database recited in the claims. During prosecution, the Examiner at first insisted that Courts taught this feature, but eventually conceded that it does not. However, the Examiner added the Dharmarajan reference as a secondary reference asserting that it taught the timing intervals recited in each of claims 1, 11, and 18 and that it was obvious to combine Courts and Dharmarajan to arrive at the present invention.

More specifically, the Examiner conceded that “Courts fails to teach the limitation further including the said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions”. However, the Examiner asserted that Dharmarajan teaches a method and apparatus for encoding session data utilized by a server computer and the use of a session timer based on the last transmission sent and that session timer being set to elapse after a predetermined amount of time. The Examiner also asserted that it is obvious to combine such teachings of Dharmarajan with Courts, etc. “because it allows for the data to be periodically written to the database”.

The Examiner is wrong on both of these assertions about Dharmarajan. Dharmarajan neither teaches that for which it has been cited nor would it have been obvious to combine Dharmarajan with Courts as suggested.



MPEP §2143 lists three requirements for a proper rejection based on obviousness, namely:

First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.

As explained below, Dharmarajan does not teach writing of the session data to the global database "at a designated time that is a function of a predetermined time interval since a last write to said data base ..." or that is "a function of at least one of (a) the number of times the HttpSession object data is updated in said local memory and (b) the number of times an http request in said http session is serviced". Hence, the rejection must fail because at least the third prong of the test is not satisfied, and without the third prong being met, the other two prongs are essentially irrelevant.

While Dharmarajan generically teaches the use of a timer for timing the transmission of data over a network at fixed intervals, the data that is transmitted in Dharmarajan has absolutely nothing to do with session data for an http session nor is it written to a shared database.

A teaching of the use of a timer for a purpose that is entirely inapposite to the use of the timer in the present invention adds nothing relevant to the teachings of Courts relevant to the present invention.

The Examiner referred to col. 13, lines 3-35 of Dharmarajan. Col. 13, lines 3-35 and Figure 10 of Dharmarajan, which concern updating of an encrypted cookie for sending to a client during a session. Applicant assumes that the Examiner is referring

to the timer referenced in connection with steps 1002 and 1004 of the flowchart of Figure 10 and the related teachings in the specification of Dharmarajan. Figure 10 and column 13 describe a technique for enhanced security during a session between a server and a client in which the server is using information obtained from a cookie received from the client. Specifically, Dharmarajan addresses a drawback of previous cookie schemes as described in col. 2, lines 39-53, quoted below:

Another drawback to using cookies stems from the fact that cookies are transmitted in the open from the client computer to the server computer. Because cookies are transmitted in the open over the Internet, there is a possibility that the cookies may be intercepted by an unauthorized recipient. An intercepted cookie may then be "replayed" by the unauthorized recipient to gain improper access to the Web server. Col. 2, lines 39-42.

To improve security, Dharmarajan discloses in connection with Figures 8-10, a technique in which, at fixed intervals, the server obtains an encrypted cookie from the client, decrypts it, identifies relevant information in the cookie, including one or more tags, uses that information during the session, and sends the client a new encrypted cookie.

The flowchart of Figure 10 comprises the details of step 920 of the flowchart of Figure 9. Step 920 relates to authenticating the cookie that it is already using during the session (col. 12, line 66 to col. 13, line 2).

As described in col. 13, lines 3-35, the server starts a session timer (step 1002) and waits for it to run out (step 1004). When it runs out, it authenticates the cookie by requesting the cookie again from the client (step 1006-1008), decoding and decrypting the cookie again (step 1010), and checking if the data is valid (step 1012). If so, it generates a new encrypted cookie and sends it to the client (step 1014) and keeps the

session going. However, if the data in the cookie is not valid (step 1012), the server ends the session (step 1018) assuming that the cookie (and therefore the session) is fraudulent. The process continuously repeats at predetermined intervals until the session is ended (see steps 1016 and 1004).

Thus, in other words, the timer at issue in Dharmarajan waits a predetermined interval between authenticating the cookie (and, if authenticated, generating a new cookie and sending it to the client).

There is no discussion in Dharmarajan about how or where the server stores any session data, let alone about when it writes locally stored session data to a shared database.

Hence, Dharmarajan does not teach “writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server system at designated times, said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions” as recited, for instance, in claim 11.

Furthermore, the Examiner already has conceded that Courts does not teach this feature. Accordingly, it would be impossible for Courts and Dharmarajan to collectively teach it.

The issue at hand is whether the references, in combination, suggest replacing the prior art scheme of Courts (i.e., updating the global http session database every time the local http session database is updated) with a scheme in which the global http session database is updated after any one of (1) an elapsed time interval, (b) a certain number of http session object data updates since the last write to the shared database,

or (c) servicing a certain number of http requests since the last write to the shared database.

A reference, such as Dharmarajan, that discloses sending a new encrypted cookie from a server to a client machine at predetermined time intervals in order to enhance security, not only does not provide this teaching, but it is utterly irrelevant. It is no more relevant than a reference that discloses intermittent windshield wipers, in which the wiper is activated to clear the windshield at predetermined time intervals.

The Examiner seems to place significant reliance on the fact that Dharmarajan is in the same field of endeavor as Courts and the present invention. However, teaching the use of a predetermined interval to perform an act inapposite to the claimed act is still irrelevant even if the inapposite act is performed by a device in the same field as the device that is claimed. By analogy, a reference that teaches utilizing compound X in windshield washer fluid that helps clean the windshield is not properly combinable with a reference teaching a power steering system to render obvious a claim to a power steering system using compound X in engine oil for the purpose of increasing the useful life of the oil simply because both references are in the art of automobiles.

Simply put, sending a new encrypted cookie to a client machine at predetermined time intervals in order to prevent unauthorized interception and use of those cookies could not possibly lead one to the realization that one should update http session data in a database shared by a server family at intervals other than the conventional interval. The two subjects have no more to do with each other, than windshield washer fluid and oil.

Thus, there is no teaching, motivation, or suggestion in the prior art of record to make the proposed combination nor do the references collectively disclose all of the claim elements.

Thus, the prior art of record does not teach or suggest to one of ordinary skill in the related arts “writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server system at designated times, said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions” as recited in claim 11 or “a second computer program adapted to write to said database a copy of said HttpSession data for each said http session at a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session” as recited in claim 1.

With respect to independent claim 18, the patentable distinctions are even more stark. Specifically, claim 18 recites different criteria for the interval between updates of the shared http session database than that recited in claims 1 and 11. Claim 18 recites that “said designated times [are] determined as a function of at least one of (a) the number of times the http session object data is updated in said local memory and (b) the number of times said http request in said http session is serviced”. Note that the “database” is previously introduced in claim 18 as being in a memory “accessible by each of said JVMs”.

Dharmarajan’s timer waits a predetermined period of time before updating the cookie. Thus, with respect to claim 18, not only does Dharmarajan disclose absolutely nothing about using a timer to update a shared http session database, but it also

discloses a completely different timing interval than claimed in any event, namely, either the number of times the http session object data is updated in said local memory of the number of times said http request in said http session is serviced. Rather, Dharmarajan's interval is a predetermined elapsed time.

Thus, the independent claims distinguish over the prior art of record because none of the references taken alone or in combination suggest updating a shared http session database based on any of the timing criteria recited in the independent claims of the present application. The core concept of the present invention is simply lacking from the prior art of record.

**B. Dependent Claims 2-10, 12-17, and 19-24**

All of the remaining claims depend from one of claims 1, 1, and 18 and therefore patentably distinguish over the prior art of record for at least al of the same reasons. Neither of the additional references cited, i.e., Prabandham and Ng , provide the aforementioned teachings lacking from Courts and Dharmarajan.

Respectfully submitted,

Dated: June 26, 2007

/Theodore Naccarella/  
Theodore Naccarella, Reg. No. 33,023  
Synnestvedt & Lechner LLP  
2600 Aramark Tower  
1101 Market Street  
Philadelphia, PA 19107  
Telephone: (215) 923-4466  
Facsimile: (215) 923-2189

TXN:pmf

**APPENDIX A: CLAIMS INVOLVED IN THIS APPEAL**

1. A server system utilizing an HttpSession object in a Java servlet application program interface (API) comprising:

a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory;

a second memory having a database for storing HttpSession objects for http sessions being handled by said JVMs, said memory being accessible by each of said JVMs;

a first computer program adapted to store in a memory local to said server running said JVM HttpSession data for each http session handled by said JVM;

a second computer program adapted to write to said database a copy of said HttpSession data for each said http session at a designated time that is a function of a predetermined time interval since a last write to said database of HttpSession object data for said http session.

2. The server system of claim 1 further comprising a third computer program adapted to write to said database a copy of said HttpSession object data for each said http session at the time the http session is initiated.

3. The server system of claim 2 wherein said plurality of JVMs are running on a plurality of servers.

4. The server system of claim 3 wherein said writes to said database are performed at the end of a corresponding servlet service method.

5. The server system of claim 4 wherein said server system services the World Wide Web.

6. The server system of claim 5 wherein said Java servlet APIs are J2EE servlet APIs.

7. The server system of claim 1 wherein said second program polls said session objects stored in said memories local to said JVMs to determine if said predetermined time interval has passed since they have been updated and wherein said second program is adapted to write to said database only copies of said HttpSession objects that have been updated within said predetermined time interval.

8. The server system of claim 7 wherein said second computer program is invoked at predetermined intervals.

9. The server system of claim 1 wherein said time interval is configurable.

10. The server system of claim 9 wherein said time interval is between ten seconds and five minutes.

11. A method of maintaining session data in a server system servicing a network, said server system maintaining state data pertaining to sessions, said method comprising the steps of:

(1) storing data for each session in a memory local to a server servicing said session;



(2) writing a copy of said data for each said session stored in said local memory into a central memory accessible to all servers of said server system at designated times, said designated times being a function of a predetermined time interval since a last write to said database of data for said sessions.

12. The method of claim 11 further comprising the step of:

(3) writing in said database a copy of said session data for each said http session at the time the http session is initiated.

13. The method of claim 11 wherein said server system services the World Wide Web.

14. The method of claim 11 wherein said server system comprises a plurality of Java Virtual Machines (JVMs) running on a plurality of servers, and wherein said data for said sessions comprises an HttpSession object of a Java servlet application program interface (API).

15. The method of claim 14 wherein said Java servlet APIs are J2EE servlet APIs.

16. The method of claim 11 wherein said time interval is configurable.

17. The method of claim 14 further comprising the step of:

(4) polling said session objects stored in said memories local to said JVMs to determine if they have been updated since the last time step (2) was performed; and

wherein, in step (2), only copies of said HttpSession objects that have been updated within said predetermined time interval are written to said database.

18. A server system utilizing HttpSession objects in a Java servlet application program interface (API) comprising:

a plurality of Java Virtual Machines (JVMs) running on at least one server, said at least one server including a local memory;

a memory having a database for storing HttpSession objects for http sessions being handled by said plurality of JVMs, said memory being accessible by each of said JVMs;

a first computer program adapted to store in a memory local to said server running said JVM HttpSession object data for each http session handled by a JVM;

a second computer program adapted to write a copy of said HttpSession data for each said http session in said database at designated times, said designated times determined as a function of at least one of (a) the number of times the HttpSession object data is updated in said local memory and (b) the number of times an http request in said http session is serviced.

19. The server system of claim 18 wherein said second computer program is adapted to write said HttpSession object data to said database after X HttpSession updates in said local memory, where X is an integer greater than or equal to 2.

20. The server system of claim 18 wherein said second computer program is adapted to write said HttpSession object data to said database after X http requests in said http sessions, where X is an integer greater than or equal to 2.

21. The server system of claim 18 further comprising a third computer program adapted to store in said database a copy of said HttpSession object data for each said http session at the time the http session is created.

22. The server system of claim 21 wherein said plurality of JVMs are running on a plurality of servers.

23. The server system of claim 22 wherein said Java servlet APIs are J2EE servlet APIs.

24. The server system of claim 18 wherein said writes to said database are performed at the end of a first servlet service method of a corresponding http session received after said designated time.

**EVIDENCE APPENDIX**

None.

**RELATED PROCEEDINGS APPENDIX**

None.